

- COMPUTAÇÃO GRÁFICA E INTERFACES -

MIEI/FCT/UNL – Ano letivo 2014/2015

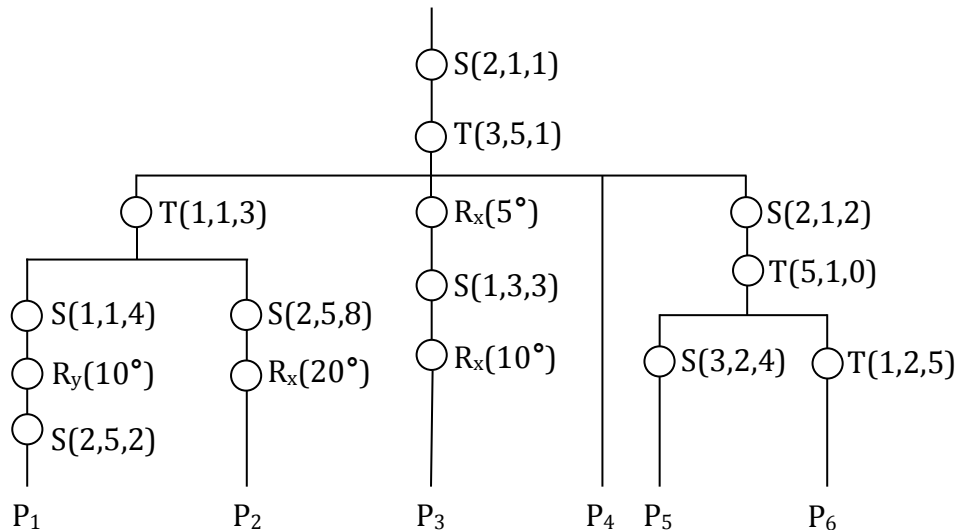
Teste 2 – 2014.12.11

Responda no próprio enunciado, que entregará. Em caso de engano, e se o espaço para as respostas não for suficiente, poderá usar o verso das folhas desde que feitas as devidas referências.

Não desagrafe as folhas! A prova, com duração de **1H30**, é **sem consulta**!

1. (4 valores)

No grafo de cena abaixo representado, os P_i representam primitivas geométricas, onde se sabe não ocorrerem quaisquer transformações geométricas.



a) Escreva, em pseudo-código, o programa OpenGL otimizado correspondente, usando o menor número de operações `glPushMatrix()` e `glPopMatrix()`:

b) Simplifique o grafo apresentado por forma a usar o menor número de nós possível e apresente, abaixo, a composição de transformações geométricas e as primitivas às quais se aplicam, apenas

Pág. 2/4 Nome _____ Número _____
para os ramos onde tenham ocorrido simplificações (por ramo entenda-se o percurso da raiz do grafo até uma dada primitiva):

Ramo 1: _____

Ramo 2: _____

Ramo 3: _____

c) Indique o número total de nós do grafo após a simplificação: _____

2. (3 valores)

a) Indique qual o nome da componente que faria variar, para uma dada cor de partida nas condições pedidas, definida no modelo **HLS**, de modo a:

- percorrer a escala de cinzentos: _____
- percorrer as cores principais do modelo RGB e CMY: _____
- passar duma cor para o seu equivalente em tons de cinzento: _____

b) Se o modelo **HSV** for representado por um cone, como descreveria o lugar geométrico (ponto, linha, triângulo, retângulo, círculo, circunferência, etc.):

- Todas as cores com máxima saturação: _____
- Todos os tons de cinzento? _____
- Todas as cores com um mesmo tom? _____

3. (2,5 valores)

Atendendo ao pipeline gráfico 3D estudado nas aulas, **indique, justificando** se:

- A iluminação poderá ou não ser efetuada após a transformação do volume de visão para o volume canónico: _____

- A transformação do volume de visão para o volume canónico, representado por um cubo, é razão suficiente para que haja deformação dos objetos na sua visualização final, num visor: _____

4. (3 valores)

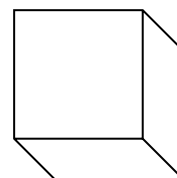
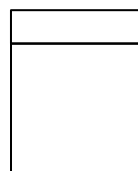
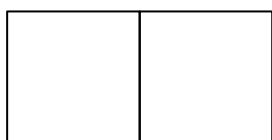
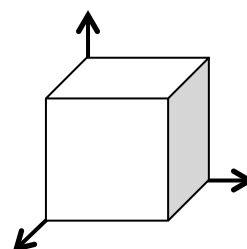
Assinale as afirmações com verdadeiro (V) ou falso (F) na quadrícula da esquerda! **Cada resposta errada descontará 50% da cotação!**

Segundo o modelo de reflexão difusa $I_{rgb} = I_{p,rgb} K_{d,rgb} \cos(\theta)$:

	a cor observada num ponto numa face plana dum objeto só depende da orientação dessa mesma face e da cor do próprio objeto
	quando o ângulo entre N e L é 90° , o objeto não refletirá qualquer luz na direção do observador
	quando $(\mathbf{N} \cdot \mathbf{L}) = 1$, assumindo que ambos os vetores estão normalizados, o objeto refletirá a totalidade da luz nele incidente
	um objeto de cor RGB(1, 0.5, 0.7) poderá ser visto como tendo a cor RGB(0.8, 0, 0.1)
	para que um objeto cinzento seja visível bastará que uma das componentes da luz não seja nula e a face esteja parcialmente virada para a luz
	Quando visível, um objeto de cor verde RGB(0,k,0), será sempre visualizado em tons de verde

5. (4.5 valores)

O cubo apresentado ao lado foi projetado recorrendo a diversas projeções planas, apresentando-se a imagem final de cada uma dessas projeções em baixo. Indique, para cada imagem, o tipo de projeção que poderá ter sido utilizado, assim como os nomes e valores dos respetivos parâmetros! Caso exista mais do que uma resposta, bastará indicar apenas uma das hipóteses.



Projeção					
Par. 1 (nome, valor)					
Par. 2 (nome, valor)					

a) Para qualquer das imagens acima, qual o número máximo de vezes que um determinado pixel da imagem poderia vir a ser pintado, assumindo que foi usado o algoritmo de z-buffer para efetuar a remoção das superfícies ocultas? _____ Justifique!

b) E no caso de se usar previamente o teste de *culling* de faces, como seria a sua resposta à alínea anterior?

6.

a) Complete o seguinte trecho dum programa em OpenGL (JOGL2), por forma a que faça sentido e que o mapeamento da textura seja o mapeamento ortogonal classico, segundo a direção do eixo y:

```
gl._____(GL_POLYGON);  
  
for(int i=0; i<nvert; i++) {  
    gl.glTexCoord2f(_____, _____);  
    gl._____(x[i], y[i], z[i]);  
}  
gl._____();
```

b) Que problema poderá surgir quando uma textura, de uma determinada resolução, é mapeada num polígono, cuja área projetada no visor (em pixels) é bastante superior à dimensão (em texels) da textura e de que forma poderá ser atenuado?

Boa sorte!